C Library for Linear Algebra Functions

Purav Parekh
Department of ME,
PES University.
Bengaluru, India
Email: purav98@gmail.com

Prof. Channa Bankapur
Dept of CSE,
PES University
Bengaluru, India
Email: channabankapur@gmail.com

***Abstract*— Linear algebra is the branch of mathematics concerning linear equations such as**

$$a_1x_1+a_2x_2+\ldots + a_nx_n=b,$$

linear functions such as

$$(x_1,\ldots x_n) \longrightarrow a_1x_1+a_2x_2+\ldots + a_nx_n\,,$$

and their representations through matrices and vector spaces. Linear algebra is central to almost all areas of mathematics.

Keywords: Matrix, Order, Elements, Gaussian, Determinant, Inverse, Linear Independence, Identity Matrix

I. INTRODUCTION

Linear algebra is fundamental in modern presentations of geometry, including for defining basic objects such as lines, planes and rotations. Also, functional analysis may be basically viewed as the application of linear algebra to spaces of functions. Linear algebra is also used in most sciences and engineering areas, because it allows modelling many natural phenomena, and efficiently computing with such models. For nonlinear systems, which cannot be modelled with linear algebra, linear algebra is often used as a first-order approximation.

A vector space over a field *F* (often the field of the real numbers) is a set *V* equipped with two binary operations satisfying the following axioms. Elements of *V* are called *vectors*, & elements of *F* are called *scalars*.

Linear maps are mappings between vector spaces that preserve the vector-space structure. Given two vector spaces *V* and *W* over a field *F*, a linear map (also called, in some contexts, linear transformation, linear mapping or linear operator) is a map

$$T: V \longrightarrow W$$

A set of vectors that spans a vector space is called a spanning set or generating set. If a spanning set *S* is *linearly dependent* (that is not linearly independent), then some element *w* of *S* is in the span of the other elements of *S*, and the span would remain the same if one remove *w* from *S*. One may continue to remove elements of *S* until getting a *linearly independent spanning set*. Such a linearly independent set that spans a vector space *V* is called a basis of *V*.

Matrices allow explicit manipulation of finite-dimensional vector spaces and linear maps. Their theory is thus an essential part of linear algebra.

Systems of linear equations form a fundamental part of linear algebra. Historically, linear algebra and matrix theory has been developed for solving such systems. In the modern presentation of linear algebra through vector spaces and matrices, many problems may be interpreted in terms of linear systems.

For example, let

$$\begin{aligned} &4x -3y = 1 \\ &2x -1y + 2z = 2 \qquad (A) \\ &x + 5y -7z = -1 \end{aligned}$$

be a linear system.

To such a system, one may associate its matrix

$$A = \begin{bmatrix} 4 & -3 & 0 \\ 2 & -1 & 2 \\ 1 & 5 & 7 \end{bmatrix}$$

and its right member vector

$$x = \begin{pmatrix} 1 \\ 2 \\ -1 \end{pmatrix}$$

The point of view and methodology explained in this paper has profound potential to make the solving of linear equations using matrices extremely quick and easy.

The identity matrix, or sometimes ambiguously called a unit matrix, of size n is the $n \times n$ square matrix with ones on the main diagonal and zeros elsewhere. It is denoted by I_n, or simply by I.

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Fig. 1: 3x3 identity matrix

In linear algebra, an n-by-n square matrix A is called invertible (also non-singular or nondegenerate) if there exists an n-by-n square matrix B such that

$$AB=BA=I$$

the matrix **B** is uniquely determined by **A** and is called the ***inverse*** of **A**, denoted by $\mathbf{A}^{-1}$.

A square matrix that is not invertible is called singular or degenerate. A square matrix is singular if and only if its determinant is 0. Singular matrices are rare in the sense that a square matrix randomly selected from a continuous uniform distribution on its entries will almost never be singular.

An **eigenvector** or **characteristic vector** of a linear transformation is a non-zero vector that changes by only a scalar factor when that linear transformation is applied to it. More formally, if T is a linear transformation from a vector space V over a field F into itself and $\mathbf{v}$ is a vector in V that is not the zero vector, then $\mathbf{v}$ is an eigenvector of T if $T(\mathbf{v})$ is a scalar multiple of $\mathbf{v}$. This condition can be written as the equation

$$T(v)=\lambda v$$

where λ is a scalar in the field F, known as the **eigenvalue**, **characteristic value**, or **characteristic root** associated with the eigenvector **v**.

II. METHODOLOGY

The methodology is to input the matrix into the system and choose the different methods of solving. The different methods/functions included are Gaussian Elimination, Echelon Form, Gauss Jordan, LU Decomposition & Rayleigh Power Method. The details of the functions are given below.

A. Echelon Form

e. $\begin{bmatrix} 1 & 2 & -3 & 4 \\ 0 & 2 & 1 & -1 \\ 0 & 0 & 1 & -3 \end{bmatrix}$ **f.** $\begin{bmatrix} 0 & 1 & 0 & 5 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 0 \end{bmatrix}$

Fig. 2: Examples of matrices in row echelon form.

A matrix is in **row echelon form** (ref) when it satisfies the following conditions.

- The first non-zero element in each row, called the **leading entry**, is 1.
- Each leading entry is in a column to the right of the leading entry in the previous row.
- Rows with all zero elements, if any, are below rows having a non-zero element.

A matrix is in **reduced row echelon form** (rref) when it satisfies the following conditions.

- The matrix satisfies conditions for a row echelon form.
- The leading entry in each row is the only non-zero entry in its column.

$$\begin{bmatrix} 1 & 2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$$

$\mathbf{A}_{\mathbf{rref}}$ $\mathbf{B}_{\mathbf{rref}}$ $\mathbf{C}_{\mathbf{rref}}$

Fig. 3: Examples of matrices in reduced row echelon form.

B. *Gaussian Elimination*

In linear algebra, Gaussian elimination (also known as row reduction) is an algorithm for solving systems of linear equations. It is usually understood as a sequence of operations performed on the corresponding matrix of coefficients. This method can also be used to find the rank of a matrix, to calculate the determinant of a matrix, and to calculate the inverse of an invertible square matrix. The method is named after Carl Friedrich Gauss (1777–1855).

To perform row reduction on a matrix, one uses a sequence of elementary row operations to modify the matrix until the lower left-hand corner of the matrix is filled with zeros, as much as possible. There are three types of elementary row operations:

- Swapping two rows,
- Multiplying a row by a nonzero number,
- Adding a multiple of one row to another row.

$$\begin{bmatrix} 0 & 1 & -3 & 4 & 1 \\ 2 & -2 & 1 & 0 & -1 \\ 2 & -1 & -2 & 4 & 0 \\ -6 & 4 & 3 & -8 & 1 \end{bmatrix} \xrightarrow{r_1 \leftrightarrow r_2} \begin{bmatrix} 2 & -2 & 1 & 0 & -1 \\ 0 & 1 & -3 & 4 & 1 \\ 2 & -1 & -2 & 4 & 0 \\ -6 & 4 & 3 & -8 & 1 \end{bmatrix}$$

$$\xrightarrow[3r_1 \text{ added to } r_4]{-r_1 \text{ added to } r_3} \begin{bmatrix} 2 & -2 & 1 & 0 & -1 \\ 0 & 1 & -3 & 4 & 1 \\ 0 & 1 & -3 & 4 & 1 \\ 0 & -2 & 6 & -8 & -2 \end{bmatrix}$$

$$\xrightarrow[2r_2 \text{ added to } r_4]{-r_2 \text{ added to } r_3} \begin{bmatrix} 2 & -2 & 1 & 0 & -1 \\ 0 & 1 & -3 & 4 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Fig. 3: The process of Gaussian Elimination including row elementary operations

TIME COMPLEXITY: The number of arithmetic operations required to perform row reduction is one way of measuring the algorithm's computational efficiency. For example, to solve a system of n equations for n unknowns by performing row operations on the matrix until it is in echelon form, and then solving for each unknown in reverse order, requires $n(n+1)/2$ divisions, $(2n^3 + 3n^2 - 5n/6)$ multiplications, and $(2n^3 + 3n^2 - 5n/6)$ subtractions, for a total of approximately $2n^3/3$ operations. Thus it has arithmetic complexity of $O(n^3)$.

C. *Gauss Jordan Method*

To find the inverse of matrix A, using Gauss-Jordan elimination, we must find a sequence of elementary row operations that reduces A to the identity and then perform the same operations on I_n to obtain A^{-1}.

For example, consider a matrix A,

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 5 & 3 \\ 1 & 0 & 8 \end{bmatrix}$$

Step 1: Adjoin the identity matrix to A,

$$\left[\begin{array}{ccc|ccc} 1 & 2 & 3 & 1 & 0 & 0 \\ 2 & 5 & 3 & 0 & 1 & 0 \\ 1 & 0 & 8 & 0 & 0 & 1 \end{array}\right]$$

Step 2: Apply row operations:

$$\left[\begin{array}{ccc|ccc} 1 & 2 & 3 & 1 & 0 & 0 \\ 2 & 5 & 3 & 0 & 1 & 0 \\ 1 & 0 & 8 & 0 & 0 & 1 \end{array}\right] \xrightarrow[R3 = R3 - \cdot R1]{R2 = R2 - 2 \cdot R1} \left[\begin{array}{ccc|ccc} 1 & 2 & 3 & 1 & 0 & 0 \\ 2-2\cdot 1 & 5-2\cdot 2 & 3-2\cdot 3 & 0-2\cdot 1 & 1-2\cdot 0 & 0-2\cdot 0 \\ 1-1 & 0-2 & 8-3 & 0-1 & 0-0 & 1-0 \end{array}\right]$$

$$\left[\begin{array}{ccc|ccc} 1 & 2 & 3 & 1 & 0 & 0 \\ 0 & 1 & -3 & -2 & 1 & 0 \\ 0 & -2 & 5 & -1 & 0 & 1 \end{array}\right] \xrightarrow[R3 = R3 + 2R2]{} \left[\begin{array}{ccc|ccc} 1 & 2 & 3 & 1 & 0 & 0 \\ 0 & 1 & -3 & -2 & 1 & 0 \\ 0+2\cdot 0 & -2+2\cdot 1 & 5+2\cdot(-3) & -1+2\cdot(-2) & 0+2\cdot 1 & 1+2\cdot 0 \end{array}\right]$$

$$\left[\begin{array}{ccc|ccc} 1 & 2 & 3 & 1 & 0 & 0 \\ 0 & 1 & -3 & -2 & 1 & 0 \\ 0 & 0 & -1 & -5 & 2 & 1 \end{array}\right] \xrightarrow[R3 = -1 \cdot R3]{} \left[\begin{array}{ccc|ccc} 1 & 2 & 3 & 1 & 0 & 0 \\ 0 & 1 & -3 & -2 & 1 & 0 \\ 0 & 0 & 1 & 5 & -2 & -1 \end{array}\right]$$

$$\left[\begin{array}{ccc|ccc} 1 & 2 & 3 & 1 & 0 & 0 \\ 0 & 1 & -3 & -2 & 1 & 0 \\ 0 & 0 & 1 & 5 & -2 & -1 \end{array}\right] \xrightarrow[R2 = R2 + 3 \cdot R3]{R1 = R1 - 3 \cdot R3} \left[\begin{array}{ccc|ccc} 1 & 2 & 0 & -14 & 6 & 3 \\ 0 & 1 & 0 & 13 & -5 & -3 \\ 0 & 0 & 1 & 5 & -2 & -1 \end{array}\right]$$

$$\left[\begin{array}{ccc|ccc} 1 & 2 & 0 & -14 & 6 & 3 \\ 0 & 1 & 0 & 13 & -5 & -3 \\ 0 & 0 & 1 & 5 & -2 & -1 \end{array}\right] \xrightarrow{R1 = R1 - 2 \cdot R2} \left[\begin{array}{ccc|ccc} 1 & 1 & 0 & -40 & 16 & 9 \\ 0 & 1 & 0 & 13 & -5 & -3 \\ 0 & 0 & 1 & 5 & -2 & -1 \end{array}\right]$$

Fig. 4: Elementary row operations to find Inverse using Gauss Jordan Method

Step 3: The inverse of A is found out:

$$A^{-1} = \begin{bmatrix} -40 & 16 & 9 \\ 13 & -5 & -3 \\ 5 & -2 & -1 \end{bmatrix}$$

The time complexity of the Gauss Jordan method is the same as that of Gauss Elimination i.e. $O(n^3)$.

D. Transpose, Rank & Orthogonality of a Matrix

The **transpose** of a matrix is a new matrix whose rows are the columns of the original one. It is denoted by A^T

Example:

$$\begin{pmatrix} 5 & 4 & 3 \\ 4 & 0 & 4 \\ 7 & 10 & 3 \end{pmatrix}^T = \begin{pmatrix} 5 & 4 & 7 \\ 4 & 0 & 10 \\ 3 & 4 & 3 \end{pmatrix}$$

Fig. 5: Example of a matrix and its transpose

The **rank** of a matrix is defined as

(a) the maximum number of linearly independent *column* vectors in the matrix

Or

(b) the maximum number of linearly independent *row* vectors in the matrix.

Example:

$$\underset{\mathbf{A}}{\begin{bmatrix} 0 & 1 & 2 \\ 1 & 2 & 1 \\ 2 & 7 & 8 \end{bmatrix}} \Rightarrow \underset{\mathbf{A_{ref}}}{\begin{bmatrix} 1 & 2 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 0 \end{bmatrix}}$$

Because the row echelon form $\mathbf{A_{ref}}$ has two non-zero rows, we know that matrix **A** has two independent row vectors; and we know that the rank of matrix **A** is 2.

An **orthogonal matrix** is a square matrix whose columns and rows are orthogonal unit vectors (i.e., orthonormal vectors), i.e.

$$QQ^T=Q^TQ=I$$

This leads to the equivalent characterization: a matrix Q is orthogonal if its transpose is equal to its inverse:

$$Q^T=Q^{-1}$$

The determinant of any orthogonal matrix is either +1 or −1.

E. Rayleigh Power Method

One of the very popular programs in C programming is Power Method, whereas a program in C can carry out the operations with short, simple and understandable codes. Power Method, used in mathematics and numerical methods, is an iteration method to compute the dominant eigenvalue and eigenvector of a matrix. It is a simple algorithm which does not compute matrix decomposition, and hence it can be used in cases of large sparse matrices. Power method gives the largest eigenvalue and it converges slowly. The algorithm is also known as the Von Mises iteration.

Although the power iteration method approximates only one eigenvalue of a matrix, it remains useful for certain computational problems. For instance, Google uses it to calculate the PageRank of documents in their search engine, and Twitter uses it to show users recommendations of who to follow.

The power iteration algorithm starts with a vector b_0, which may be an approximation to the dominant eigenvector or a random vector. The method is described by the recurrence relation

$$b_{k+1}=Ab_k/||Ab_k||$$

So, at every iteration, the vector b_k is multiplied by A the matrix and normalized.

If we assume A has an eigenvalue that is strictly greater in magnitude than its other eigenvalues and the starting vector b_0 has a nonzero component in the direction of an eigenvector associated with the dominant eigenvalue, then a subsequence (b_k) converges to an eigenvector associated with the dominant eigenvalue.

F. LU Decomposition

In numerical analysis and linear algebra, LU decomposition (where 'LU' stands for 'lower upper', and also called LU factorization) factors a matrix as the product of a lower triangular matrix and an upper triangular matrix. Computers usually solve square systems of linear equations using the LU decomposition, and it is also a key step when inverting a matrix, or computing the determinant of a matrix.

Let A be a square matrix. An LU factorization refers to the factorization of A, with proper row and/or column orderings or permutations, into two factors, a lower triangular matrix L and an upper triangular matrix U, **A=LU**.

$$\begin{bmatrix} A00 & A01 & A02 \\ A10 & A11 & A12 \\ A20 & A21 & A22 \end{bmatrix} = \overbrace{\begin{bmatrix} 1 & 0 & 0 \\ L10 & 1 & 0 \\ L20 & L21 & 1 \end{bmatrix}}^{\text{Lower Triangular}} \underbrace{\begin{bmatrix} U00 & U01 & U02 \\ 0 & U11 & U12 \\ 0 & 0 & U22 \end{bmatrix}}_{\text{Upper Triangular}}$$

Fig. 6: LU Decomposition of a matrix

Doolittle Algorithm:

It is always possible to factor a square matrix into a lower triangular matrix and an upper triangular matrix. That is, **[A] = [L][U]**

Doolittle's method provides an alternative way to factor A into LU decomposition without going through the hassle of Gaussian Elimination.

For a general n×n matrix A, we assume that an LU decomposition exists, and write the form of L and U explicitly. We then systematically solve for the entries in L and U from the equations that result from the multiplications necessary for A=LU.

For each i = 0, 1, 2 . . . . n-1 :

$$U_{i,k} = A_{i,k} - \sum_{j=0}^{i} (L_{i,j}\, U_{j,k})$$

for k=i,i+1...n-1 produces the kth row of U.

$$L_{i,k} = \left(A_{i,k} - \sum_{j=0}^{i} (L_{i,j}\, U_{j,k}) \right) / U_{k,k}$$

for i=k+1,k+2,....n-1 and $L_{i,i} = 1$ produces the k^{th} column of L

Fig. 7: Doolittle Algorithm

The time complexity of the Doolittle Algoritm is $O(n^3)$.

III. RESULTS AND DISCUSSION

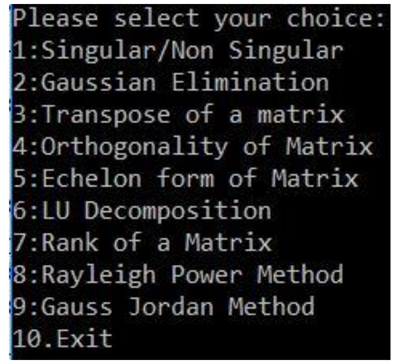

Fig 7: Choosing the operations to carry out

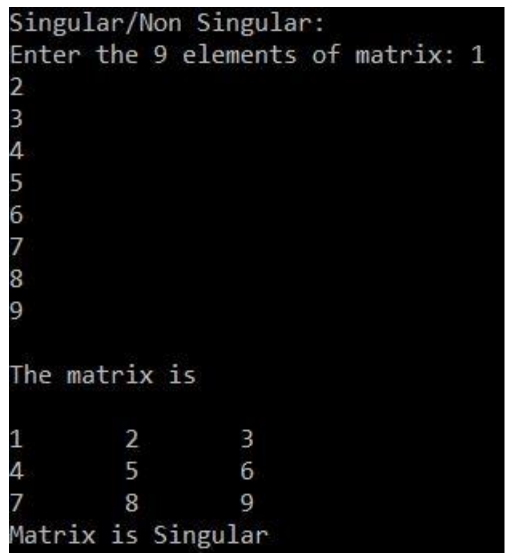

Fig 8: Finding out Singularity of a matrix

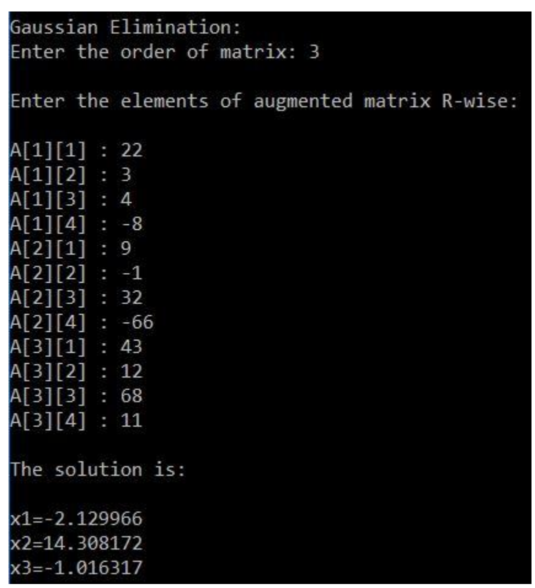

Fig 9: Carrying out Gaussian Elimination

```
Transpose:
Enter the number of rows and columns of matrix
2
3
Enter elements of the matrix
1
2
3
4
5
6
Transpose of the matrix:
1       4
2       5
3       6
```

Fig 10:Transposing A Matrix

```
Echelon form of Matrix:
Enter order of the matrix:3
Enter the matrix of order 3x3
4
-9
8
9
-6
3
-2
4
1
4.000    -9.000   8.000
0.000    57.000   -60.000
0.000    0.000    1020.000
```

Fig 11: Reducing the matrix into Echelon Form

```
LU Decomposition:
Lower Triangular                        Upper Triangular
1       0       0                       2       -1      -2
-2      1       0                       0       4       -1
-2      -1      1                       0       0       3
```

Fig 12: LU Decomposition

```
Rank of a Matrix
Intermediate Steps
 1 2 1
 0 3 1
 0 5 6

Intermediate Steps
 1 0 0
 0 3 1
 0 0 4

Intermediate Steps
 1 0 0
 0 3 0
 0 0 4

Rank of the matrix is : 3
```

Fig 13: Finding out the rank of matrix also showing the intermediate steps of reduction to reduced row echelon form

```
Rayleigh Power Method
Enter the order of matrix:3

Enter matrix elements R-wise
A[1][1]=21
A[1][2]=43
A[1][3]=67
A[2][1]=-9
A[2][2]=-23
A[2][3]=11
A[3][1]=56
A[3][2]=17
A[3][3]=29

Enter the Cumn vector
X[1]=1
X[2]=1
X[3]=1

 The required eigen value is 86.810242

The required eigen vector is :
1.000000        0.015515        0.972932
```

Fig 14: Finding the Eigen value and the Eigen vector of a matrix

```
Gauss Jordan Method
Enter order of matrix: 4
Enter the matrix:
32
45
99
-67
32
65
71
-93
13
12
37
11
23
12
67
90
The inverse matrix is:
-0.29   -0.01   1.54    -0.41
0.07    0.04    -0.55   0.16
0.08    -0.01   -0.31   0.08
0.01    0.01    -0.09   0.03
```

Fig 15: Finding the Inverse of a matrix using Von Mises iteration

IV. CONCLUSION

Linear algebra is vital in multiple areas of science in general. Because linear equations are so easy to solve, practically every area of modern science contains models where equations are approximated by linear equations (using Taylor expansion arguments) and solving for the system helps the theory develop.

Finding faster algorithms will help solve the numerous huge problems on matrices and linear equations with ease.

V. REFERENCES

[1] P R Kolhe, M H Tharkar, Pradip Kolhe, S Gawande Dr BSKKV Dapoli, “ICT Tool: - ‘C’ Language Program for Power Method” in International Journal of Emerging Technologies in Engineering Research (IJETER) Volume 5, Issue 6, June (2017)